ＬＳＩの設計方法および検証方法

発明の背景

本発明は、ＬＳＩの設計および検証に関するものであり、特に、設計データの機密を保つための技術に属する。

ＬＳＩの設計において、回路の中身の機密を保ちたい場合がある。例えば、データの暗号化に関わるＬＳＩでは、その回路の中身が知られると、これにより、暗号化のロジックが解読されてしまうおそれがある。

従来では、このような機密保持を要するＬＳＩの設計は、設計に従事する人を限定したり、あるいは、設計を行う場所を特定したりすることによって、他の人に回路の中身が知られないようにしていた。

近年のＬＳＩの複雑化、大規模化に伴い、１個のＬＳＩの設計には、多数の設計者が携わっている。したがって、設計者や設計場所の限定のみでは、回路の機密を保つのに必ずしも十分ではない。

発明の概要

本発明は、ＬＳＩの設計において、暗号化処理を採り入れ、従来よりも回路設計データの機密性を高めることを目的とする。

また、暗号化された設計データについて、機密性を保ったまま、検証を実行できるようにすることを目的とする。

具体的には本発明は、ＬＳＩ設計方法として、与えられた回路設計データに対し、暗号化処理を行うステップを備えたものである。

そして、前記本発明に係るＬＳＩ設計方法における暗号化ステップは、前記回路設計データが表す回路の全体または一部を元の回路として回路変換を行い、暗号化回路を生成するステップを備えており、前記回路変換ステップは、前記元の回路と入力数および出力数が同一の少なくとも１つのダミー回路を前記元の回路と並列に配置し、前記元の回路およびダミー回路の出力を並び替える並び替え回路を配置し、前記並び替え回路の出力から、選択信号に従って、前記元の回路の出力数に相当する数の信号を選択するセレクタを配置し、前記暗号化回路を生成

するものとし、かつ、前記選択信号をキー信号とし、前記元の回路の出力と前記セレクタの出力とが一致するようなキー信号の値を前記暗号化回路のキーとするものであるのが好ましい。

さらに、前記暗号化ステップは、前記回路変換ステップにおいて用いるダミー回路を生成するステップを備え、前記ダミー回路生成ステップは、前記元の回路に対し変換ルールに従ってダミー回路の候補からなるダミー論理データベースを生成し、前記ダミー論理データベースから出力ルールに従ってダミー回路を選択するものであるのが好ましい。また、前記変換ルールは、論理値の反転、論理演算子の変換、および論理演算子の順序変更のうちの少なくとも１つを含むのが好ましい。また、前記出力ルールは、ランダム選択であるのが好ましい。

また、前記暗号化回路のレイアウトを行うステップを備え、前記レイアウトステップは、前記キー信号の入力信号線を電源およびグランドのいずれにも接続可能なようにレイアウトを行うのが好ましい。さらに、前記レイアウトにおいて、前記キーにしたがって前記キー信号を電源およびグランドのいずれか一方に接続し、前記元の回路のレイアウトを生成するステップを備えるのが好ましい。

また、具体的には本発明は、ＬＳＩの検証方法として、リファレンスとなる動作モデルとともに暗号化された回路設計データについて、回路動作の検証を行うステップを備え、前記検証ステップは、前記暗号化された回路設計データを復号化し、実際の設計データと前記動作モデルとを得るステップと、前記実際の設計データについてシミュレーションを実行し、実出力値を得るステップと、前記動作モデルについてシミュレーションを実行し、出力期待値を得るステップと、前記実出力値と前記出力期待値とを比較し、比較結果を出力するステップとを備えたものである。

また、具体的には本発明は、ＬＳＩの検証方法として、プロトコル定義とともに暗号化された回路設計データについて、回路動作の検証を行うステップを備え、前記検証ステップは、前記暗号化された回路設計データを復号化し、実際の設計データと前記プロトコル定義とを得るステップと、前記実際の設計データについてシミュレーションを実行し、実出力値を得るステップと、前記実出力値を前記プロトコル定義と比較し、比較結果を出力するステップとを備えたものであ

る。

また、具体的には本発明は、ＬＳＩの検証方法として、暗号化された回路設計データをシミュレーションによって検証するステップを備え、前記検証ステップは、不正なアクセスによるシミュレーションを制限するものである。

そして、前記検証ステップは、前記暗号化された回路設計データを復号化し、実際の設計データを得るステップと、前記実際の設計データについてシミュレーションを実行するステップと、前記シミュレーションにおいて、所定の制限情報をカウントするステップと、カウント値が上限値を超えたとき、シミュレーションに制限措置を施すステップとを備えているのが好ましい。さらに、前記所定の制限情報は、シミュレーションの実行ステップ、シミュレーションの実行時間、回路内の特定信号のトグル数、および回路への入力の組み合わせのうちの少なくとも１つを含むのが好ましい。さらに、所定の制限情報をランダムに選択するのが好ましい。

また、前記検証ステップは、前記暗号化された回路設計データを復号化し、実際の設計データを得るステップと、前記実際の設計データについてシミュレーションを実行するステップと、前記シミュレーションにおいて、所定のプロトコル制約条件を違反するか否かをチェックするステップと、違反したとき、シミュレーションに制限措置を施すステップとを備えているのが好ましい。そして、前記所定のプロトコル制約条件は、入力プロトコル、および動作中プロトコルのうちの少なくとも１つを含むのが好ましい。さらに、前記所定のプロトコル制約条件をランダムに選択するのが好ましい。

さらに、前記本発明に係るＬＳＩ検証方法における制限措置は、シミュレーションの停止、実行速度低下および異常実行、シミュレーション結果の非出力、並びに、次ステップに渡すデータまたはキーの非生成のうちの少なくとも１つを含むのが好ましい。

また、具体的には本発明は、ＬＳＩ検証方法として、シミュレーションにおける不正アクセスをチェックするチェック回路を含む回路設計データを、暗号化処理するステップと、暗号化された回路設計データを、シミュレーションによって検証するステップとを備え、前記検証ステップは、前記チェック回路を動作させ

て、不正なアクセスによるシミュレーションを制限するものである。

そして、前記チェック回路は、シミュレーションにおいて、所定の制限情報のカウント値が上限値を超えたか否かをチェックするのが好ましい。また、前記チェック回路は、シミュレーションにおいて、プロトコル制約条件の違反の有無をチェックするのが好ましい。

また、具体的には本発明は、ＬＳＩ設計方法として、与えられた回路設計データから、タイミング情報を抽出するステップと、前記回路設計データを抽出したタイミング情報のみを合わせて所定の変換ルールに従い、暗号化設計データに変換するとともに、少なくとも１つの論理ゲートにバッファを付加するステップと、前記暗号化設計データについて付加したバッファのサイズを調整するステップと、バッファサイズ調整後の前記暗号化設計データを、前記所定の変換ルールをキーとして復号化するステップとを備えたものである。

また、具体的には本発明は、ＬＳＩ設計方法として、固有ＩＤの判定回路とともに暗号化された回路設計データについて復号化を行い、実際の設計データと前記固有ＩＤの判定回路とを得るステップを備え、前記ステップは、入力された固有パラメータによって前記固有ＩＤの判定回路に正しい値を定義するステップを有するものである。

図面の簡単な説明

図１は本願発明者が提案する回路設計スタイルを示す図である。

図２は図１の回路設計スタイルにおける基本工程のパターンを示す図である。

図３（ａ），（ｂ）は本発明に係る処理の流れを示す図である。

図４は暗号化処理の一例としての回路変換を示す図である。

図５（ａ）～（ｄ）は図４に示す回路変換の具体例を示す図である。

図６は図４に示す回路変換に用いるダミー回路の生成方法を示す図である。

図７（ａ），（ｂ）は復号化処理を説明するための図である。

図８は検証処理としての第１の判定方法を示す図である。

図９は検証処理としての第２の判定方法を示す図である。

図１０は検証処理としてのシミュレーション制限方法を示す図である。

図１１（ａ），（ｂ）はチェック回路の一例を示す図である。

図１２（ａ），（ｂ）はチェック回路の一例を示す図である。

図１３はタイミング調整方法を示す図である。

図１４はタイミング調整方法を示す図である。

図１５は暗号化設計データが表す回路構成の一例を示す図である。

図１６は暗号化設計データが表す回路構成の一例を示す図である。

発明の詳細な説明

図１は本願発明者が提案する回路設計スタイルを示す図である。図１に示す設計スタイルでは、機密設計データの中身が見えなくても設計・検証処理ＳＢが実行可能なように、設計データの暗号化処理ＳＡおよび復号化処理ＳＣが実行される。

暗号化処理ＳＡでは、機密を必要とする回路の設計データ１１に対して暗号化を行い、暗号化設計データ１２と、この暗号化を解除するキー１３とを生成する。暗号化設計データ１２は設計・検証処理Ｓ２を実行する利用者に提供される。また、その設計・検証処理Ｓ２の必要に応じて、キー１３も併せて提供される。

設計・検証処理ＳＢでは、暗号化設計データ１２について、元の回路の中身が開示されることなく、各種の処理が行われる。復号化処理ＳＣでは、設計・検証処理Ｓ２が実行された後の暗号化設計データ１４に対してキー１５を用いて復号化を行い、元の回路の設計データ１６を生成する。

図２は図１の回路設計スタイルにおける基本工程のパターンを示す図である。同図中、（ａ）は暗号化Ａ、（ｂ）は暗号データのままの処理Ｂ１、（ｃ）は暗号を保ったデータ変換Ｂ２、（ｄ）は復号化Ｃを示している。処理Ｂ１と処理Ｂ２とは、処理Ｂ１では復号化および暗号化は行われず、新たなキーは生成されないが、処理Ｂ２では新たな暗号化データが新たなキーとともに生成される点で相違する。

図３は本発明に係る処理の流れを示す図であり、図２に示す基本工程のパターンを組み合わせたものである。同図中、（ａ）に示す処理は、暗号化Ａ、暗号デ

ータのままの処理Ｂ１および復号化Ｃを組み合わせたものである。例えば、ＲＴＬレベルまたはビヘイビアレベルの設計データを暗号化し、この暗号化データについて論理合成を行い、暗号化されたゲートレベルの設計データを出力し、その後ゲートレベルの設計データに復号化する、といった処理がこれに相当する。これにより、論理合成中の設計データの機密を保つことができる。また、暗号化データについて論理合成およびレイアウトを行い、その後マスクデータを復号化するような場合も考えられる。また、（ｂ）に示す処理は、暗号化Ａ、暗号を保ったデータ変換Ｂ２および復号化Ｃを組み合わせたものである。

以下、各処理の具体例について、順に説明する。

＜暗号化処理＞

（回路変換）

図４は本発明に係る暗号化処理の一例である回路変換を示す図である。図４において、ｆ０は元の暗号化されていない回路である。回路ｆ０の入力数をｎ、出力数をｍとする。回路ｆ０は元の回路の全体を表すものであってもよいし、元の回路の部分回路であってもかまわない。

図４に示すように、回路ｆ０と並列に、回路ｆ０と入力数、出力数が同一の（ｐ－１）個のダミー回路ｆ１～ｆｐ－１を配置する。そして、その後段に、並び替え回路２１およびセレクタ２２を設ける。並べ替え回路２１は回路ｆ０の出力と各ダミー回路ｆ１～ｆｐ－１の出力とを受け、これらの出力を並べ替えて出力する。例えば出力Ｏ１からは、各回路ｆ０～ｆｐ－１の出力の第１ビットを集め並び替えて出力し、出力Ｏ２からは、各回路ｆ０～ｆｐ－１の出力の第２ビットを集め並び替えて出力する。これにより、並べ替え回路２１から、元の回路ｆ０の出力数に相当する個数すなわちｍ個の、ｐビットの信号が出力される。

セレクタ２２は、選択信号ＫＥＹに従って、並び替え回路２１の各出力から１ビットずつを選択し、出力する。これにより、回路ｆ０と同じｍ個の信号がセレクタ２２から出力される。このような回路変換の結果、図４に示すような暗号化回路が生成される。

ここで、選択信号ＫＥＹを暗号化回路のキー信号とする。そして、回路ｆ０の出力とセレクタ２２の出力とが一致するようなキー信号ＫＥＹの値を、暗号化回

路のキーとする。

このような回路変換による暗号化は、変換手順が簡易であり、自動変換が容易である。また、暗号化による遅延増加は、セレクタ２２における遅延のみであり、極めて少ない。

図５は図４に示す回路変換の具体例を示す図である。いま、元の回路ｆ０として、図５（ａ）に示すような２入力２出力の回路が与えられたとする。この回路ｆ０に対して図５（ｂ）に示すようなダミー回路ｆ１を配置する。図５（ｃ）は、回路ｆ０をダミー回路ｆ１を用いて暗号化した結果を示す図である。さらに、図５（ｃ）の回路を合成して、図５（ｄ）のような暗号化回路を得る。この回路のキーは（０，１）である。

図６は図４に示す回路変換に用いるダミー回路の生成方法を示す図である。図６に示すように、元の回路ｆ０に対して、所定の変換ルール２５に従って、ダミー回路候補を含むダミー論理データベース（ＤＢ）２６を生成する。そして、生成したダミー論理ＤＢ２６から、所定の出力ルール２７に従って、任意のダミー回路ｆ１～ｆｐ－１を出力する。このような生成方法では、変換ルール２５および出力ルール２７の設定によって、ダミー回路ｆ１～ｆｐ－１を柔軟に生成することができるので、自動化処理に適する。

変換ルール２５の例としては、論理値の反転、論理演算子の変換、論理演算子の順序変更などが挙げられる。論理値の反転では、入力値の反転や出力値の反転の他に、複数ビット信号の一部のビットを反転する方法が考えられる。論理演算子の変換では、ＡＮＤとＯＲとの変換が考えられる。また、出力ルール２７の例としては、ランダムに選択する方法や、重複するダミー回路を排除する方法などが考えられる。

＜復号化処理＞

図４に示すような回路変換によって得られた暗号化回路をレイアウトする際に、キー信号の入力信号線を、電源およびグランドのいずれにも接続可能なようにレイアウトを行う。これによって、レイアウト工程まで、元の回路の内容について機密を保つことができ、かつ、キーを用いて極めて容易に元の回路の復号化を実現することができる。

図７は本復号化処理を説明するための図であり、同図中、（ａ）は暗号化回路のレイアウトの一例、（ｂ）は（ａ）の回路をキーに従って復号化した結果を示す図である。図７（ａ）に示すように、暗号化回路３０に入力されるキー信号ＫＥＹの入力信号線３１を、電源ＶＤＤおよびグランドＶＳＳのいずれにも接続可能なように、レイアウトを行う。そして、図７（ｂ）に示すように、キー（図の例では（０，１，０））に従って、キー信号ＫＥＹの入力信号線３１を電源およびグランドのいずれか一方に接続する（ＥＣＯ（Engineering Change Order））。これによって、元の回路のレイアウトが復号化される。

＜設計・検証処理＞

（判定）

暗号化設計データを、復号化してシミュレーションによる検証を行うとき、シミュレーション結果が正常であるか否かを判定するための期待値が必要になる。ところが、この期待値が外部から見えたとすると、回路の内容がこの期待値から推定可能になり、設計データの機密性が保てない。

そこでここでは、回路設計データを暗号化するとき、シミュレーション結果の期待値となるデータまたは期待値を得る元になるデータを含めて、暗号化を行う。そして、検証処理では、シミュレーション結果と期待値との比較結果に基づき、回路動作が正常か否かを判定する。

図８は本発明に係る検証方法としての第１の判定方法を示す図である。本判定方法では、シミュレーション結果の期待値を得る元になるデータとして、リファレンスとなる動作モデルを用いる。すなわち、図８に示すように、まず、動作モデルとともに暗号化された回路設計データ４１をキー５２を用いて復号化し（Ｓ２１）、実際の下位レベルの設計データ４３（ＲＴＬまたはゲートレベルのネットリスト）と動作モデルの設計データ４４とを得る。そして、下位レベルの設計データ４３についてシミュレーションを実行し（Ｓ２２）、実出力値４５を得る。また、動作モデルの設計データ４４についてシミュレーションを実行し（Ｓ２３）、出力期待値４６を得る。そして、得られた実出力値４５と出力期待値４６を比較し、各シミュレーション時間において、実出力値４５と出力期待値４６とが一致しているか否かを判定する（Ｓ２４）。図８の例では、値が一致してい

るので、結果通知４７として、シミュレーション結果は正常である旨を出力する。

図９は本発明に係る検証方法としての第２の判定方法を示す図である。本判定方法では、シミュレーション結果の期待値となるデータとして、プロトコル定義を用いる。すなわち、図９に示すように、まず、プロトコル定義とともに暗号化された回路設計データ５１をキー５２を用いて復号化し（Ｓ３１）、設計データ５３と、プロトコル定義５４とを得る。プロトコル定義５４では、設計データ５３が示す回路の入出力および中間ノードの値について、その動作状態が定義されている。そして、設計データ５３についてシミュレーションを実行し（Ｓ３２）、得られた実出力値５５と、プロトコル定義５４とを比較する（Ｓ３３）。

なお、第１および第２の判定方法において、シミュレーション結果が異常である旨の結果が得られたときは、シミュレーション実行結果である実出力値４５，５５を、暗号化して出力してもよい。

（シミュレーション制限）

暗号化設計データについてシミュレーションを実行し、検証した場合、検証結果出力には、設計データ内の全ての信号線の情報が格納される。多くの入力を与えてシミュレーションを実行し、それから得た検証結果出力を解析すれば、暗号化された回路の内容を知得することが可能になる。

そこで、ここでは、検証結果出力から回路の内容が知得されないように、言い換えると、不正アクセスを監視、防止するために、シミュレーションに制限をかける方法を示す。

図１０は本発明に係る検証方法としてのシミュレーション制限方法を示す図である。図１０に示すように、シミュレーションＳ４２のチェックＳ４３では、次のような所定の制限情報６４を、シミュレーションの間カウントする。そして、カウント値が所定の上限値を越えたとき、シミュレーションＳ４２に制限措置を施す。

・シミュレーションの実行ステップ、実行時間

・回路内の特定信号のトグル数

・回路への入力の組み合わせ

これらの制限情報を、ランダムに選択するようにしてもよい。また、シミュレーションの制限措置としては、次のようなものが考えられる。

・シミュレーションの停止、実行速度低下、異常実行

・シミュレーション結果の非出力

・各信号線のダンプ情報、判定結果などのデータの出力停止

・次ステップに渡すデータまたはキーの非生成

また、制限情報６４として、プロトコルの制約条件を設けて、シミュレーションにおいて、このプロトコル制約条件に違反するか否かを判定してもよい。プロトコルの制約条件としては、次のようなものが挙げられる。

・回路への入力において許容できるプロトコル（入力プロトコル）

・回路内の動作において許容できるプロトコル（動作中プロトコル）

これらのプロトコル制約条件を、ランダムに選択するようにしてもよい。

なお、制限情報６４は、回路設計データの暗号化の際に併せて暗号化し、復号化してもかまわないし、暗号化設計データ６１とは別に与えてもよい。

また、シミュレーションにおいて不正アクセスをチェックするための回路を予め回路設計データに含めておいて、暗号化してもよい。このチェック回路は、シミュレーションのときにのみ動作し、回路設計後は動作がディセイブルされるように、構成する必要がある。

図１１はシミュレーションにおいて不正アクセスをチェックする回路の一例を示す図である。図１１（ａ）に示すチェック回路は、信号Ａの変化回数が所定値（ここでは「８」）を越えたとき、信号Ｂの値にかかわらず、出力値を“０”に固定するものである。シミュレーション時には、信号Ｘに“１”→“０”を与え、外部リセットをかける。その後、図１１（ｂ）に示すように、所定の制限情報としての信号Ａの変化回数が８を越えるまでは出力値は信号Ｂの値に一致するが、信号Ａの変化回数が８を越えると、出力値は“０”に固定される。これにより、正しいシミュレーション結果が得られなくなる。回路製造時は、信号Ｘを“１”に固定し、このチェック回路が動作しないようにする。

図１２はチェック回路の他の例を示す図である。図１２（ａ）に示すチェック回路は、信号Ｙの変化時において信号Ａ，Ｂがともに“０”であることをプロト

コル制約条件とし、信号Ｙの変化時に信号Ａ，Ｂのいずれかが“０”でないときはプロトコル違反として認識し、信号Ｃの値にかかわらず、出力値を“０”に固定するものである。シミュレーション時には、信号Ｘに“１”→“０”を与え、外部リセットをかける。その後、図１２（ｂ）に示すように、信号Ｙの変化時において、信号Ａと信号Ｂがともに“０”でないときは、出力値は“０”に固定される。これにより、正しいシミュレーション結果が得られなくなる。回路製造時は、信号Ｘを“１”、信号Ｙを“０”に固定し、このチェック回路が動作しないようにする。

（タイミング調整）

図１３および図１４は本発明に係るＬＳＩ設計方法としてのタイミング調整方法を示す図である。このタイミング調整方法は、図３（ａ）に示すフローすなわち、暗号化Ａ→処理Ｂ１→復号化Ｃの流れを含むものである。

図１３に示すように、まず、元の回路設計データ７１からタイミング情報７２を抽出する（Ｓ５１）。ここでのタイミング情報とは、各論理ゲートから接続先の負荷までの遅延のことをいう。そして、所定の変換ルール７３に従って、元の回路の内容が分からないように、かつ、タイミング情報７２のみを合わせるように、すなわち各論理ゲートから接続先の負荷までの遅延を変化させないで、暗号化処理を行い（Ｓ５２）、暗号化設計データ７４を生成する。変換ルール７３の内容がそのままキー７５になる。このとき、図１４（ａ），（ｂ）に示すように、変換後の論理ゲート７８の少なくとも１つについて、タイミング調整のためのバッファ７９を付加する。

そして、暗号化設計データ７４について、図１４（ｃ）に示すように、目標タイミングを満たすように、付加したバッファ７９のサイズを調整する（Ｓ５３）。ここでは、暗号化処理Ｓ５２で生成したキー７５を用いる必要はない。バッファサイズ調整後、キー７５すなわち所定の変換ルールを用いて、元の回路７７を復号化する（Ｓ５４）。このとき、図１４（ｄ）に示すように、調整されたバッファサイズを基にして、元の論理ゲートに変換する。この結果、目標タイミングが達成される。

このようなタイミング調整方法によると、設計者に対して回路内容を秘匿した

まま、タイミング調整を行わせることができる。

（固有ＩＤ生成）

図１５は暗号化設計データが表す回路構成の一例を示す図である。図１５に示すように、暗号化設計データは、製品の種別を判定するための固有ＩＤの判定回路としての回路固有ＩＤレジスタを持っている。この回路固有ＩＤレジスタに入力される固有ＩＤの値は、変数で定義されている。また、この固有ＩＤの変数の値を与える固有パラメータが定義されている。固有パラメータは、実際の固有ＩＤとは異なる並び順で、定義される。

シミュレーションを行う際に、入力する固有ＩＤの値を“１１０”としても、回路固有ＩＤレジスタの値とは並びが異なっており、“１０１”と新たに定義された固有ＩＤの値が回路固有ＩＤと一致して、はじめて、正常に動作する回路に生成される。また、次の設計工程では、入力する固有ＩＤの値を“０１１”としても、新たに“１０１”と定義しなおされる。

なお、ここでの説明では、固有ＩＤおよび入力する固有ＩＤの値のビット数を３としたが、この値は任意とする。さらに、入力する固有ＩＤと回路固有ＩＤレジスタは、並びだけではなく、論理が反転していてもよい。

また、図１６に示す回路では、固有ＩＤのみではなく、論理回路中の、電源またはグランドに固定された全てのノードを全て変数で定義されている。

変数の値は、入力する固有ＩＤの値と、それ以外の正常に動作するように固定された値が入力され、回路上では、回路固有ＩＤレジスタ（図中のＡ，Ｂ，Ｃ）とそれ以外のレジスタ（図中のＤ，Ｅ）との区別がつかなくなり、容易に固有ＩＤの値を知ることができなくなる。

なお、上述した各方法は、当該方法を実現するためのプログラムを実行するコンピュータを備えた装置によって実現することができる。また、当該方法を実現するためのプログラムをコンピュータ読み取り可能な記録媒体に記録して、この記録媒体に記録したプログラムをコンピュータに実行させることによって実現することができる。

以上のように本発明によると、暗号化によって、従来よりも回路設計データの機密性を高めることができる。また、暗号化された回路設計データを、機密を保

持したまま、設計・検証させることができる。

クレーム

１．　ＬＳＩ設計方法は、

与えられた回路設計データに対し、暗号化処理を行うステップを備えている。

２．　クレーム１のＬＳＩ設計方法において、

前記暗号化ステップは、

前記回路設計データが表す回路の全体または一部を元の回路として、回路変換を行い、暗号化回路を生成するステップを備えており、

前記回路変換ステップは、

前記元の回路と入力数および出力数が同一の少なくとも１つのダミー回路を、前記元の回路と並列に配置し、

前記元の回路およびダミー回路の出力を並び替える並び替え回路を配置し、

前記並び替え回路の出力から、選択信号に従って、前記元の回路の出力数に相当する数の信号を選択するセレクタを配置し、前記暗号化回路を生成するものであり、かつ、

前記選択信号をキー信号とし、前記元の回路の出力と前記セレクタの出力とが一致するようなキー信号の値を、前記暗号化回路のキーとする。

３．　クレーム２のＬＳＩ設計方法において、

前記暗号化ステップは、前記回路変換ステップにおいて用いるダミー回路を生成するステップを備え、

前記ダミー回路生成ステップは、

前記元の回路に対し、変換ルールに従って、ダミー回路の候補からなるダミー論理データベースを生成し、

前記ダミー論理データベースから、出力ルールに従って、ダミー回路を選択する。

４．　クレーム３のＬＳＩ設計方法において、

前記変換ルールは、論理値の反転、論理演算子の変換、および論理演算子の順

序変更のうちの少なくとも１つを含む。

５．　クレーム３のＬＳＩ設計方法において、

前記出力ルールは、ランダム選択である。

６．　クレーム２のＬＳＩ設計方法において、

前記暗号化回路のレイアウトを行うステップを備え、

前記レイアウトステップは、

前記キー信号の入力信号線を、電源およびグランドのいずれにも接続可能なように、レイアウトを行う。

７．　クレーム６のＬＳＩ設計方法において、

前記レイアウトにおいて、前記キーにしたがって前記キー信号を電源およびグランドのいずれか一方に接続し、前記元の回路のレイアウトを生成するステップを備えている。

８．　ＬＳＩの検証方法は、

リファレンスとなる動作モデルとともに暗号化された回路設計データについて、回路動作の検証を行うステップを備え、

前記検証ステップは、

前記暗号化された回路設計データを復号化し、実際の設計データと、前記動作モデルとを得るステップと、

前記実際の設計データについてシミュレーションを実行し、実出力値を得るステップと、

前記動作モデルについてシミュレーションを実行し、出力期待値を得るステップと、

前記実出力値と前記出力期待値とを比較し、比較結果を出力するステップとを備えている。

９．　ＬＳＩの検証方法は、

プロトコル定義とともに暗号化された回路設計データについて、回路動作の検証を行うステップを備え、

前記検証ステップは、

前記暗号化された回路設計データを復号化し、実際の設計データと、前記プロトコル定義とを得るステップと、

前記実際の設計データについてシミュレーションを実行し、実出力値を得るステップと、

前記実出力値を、前記プロトコル定義と比較し、比較結果を出力するステップとを備えている。

１０．　ＬＳＩの検証方法は、

暗号化された回路設計データを、シミュレーションによって検証するステップを備え、

前記検証ステップは、不正なアクセスによるシミュレーションを制限する。

１１．　クレーム１０のＬＳＩ検証方法において、

前記検証ステップは、

前記暗号化された回路設計データを復号化し、実際の設計データを得るステップと、

前記実際の設計データについてシミュレーションを実行するステップと、

前記シミュレーションにおいて、所定の制限情報をカウントするステップと、

カウント値が上限値を超えたとき、シミュレーションに制限措置を施すステップとを備えている。

１２．　クレーム１１のＬＳＩ検証方法において、

前記所定の制限情報は、シミュレーションの実行ステップ、シミュレーションの実行時間、回路内の特定信号のトグル数、および回路への入力の組み合わせのうちの少なくとも１つを含む。

１３．　クレーム１２のＬＳＩ検証方法において、

前記所定の制限情報を、ランダムに選択する。

１４．　クレーム１０のＬＳＩ検証方法において、

前記検証ステップは、

前記暗号化された回路設計データを復号化し、実際の設計データを得るステップと、

前記実際の設計データについてシミュレーションを実行するステップと、

前記シミュレーションにおいて、所定のプロトコル制約条件を違反するか否かをチェックするステップと、

違反したとき、シミュレーションに制限措置を施すステップとを備えている。

１５．　クレーム１４のＬＳＩ検証方法において、

前記所定のプロトコル制約条件は、入力プロトコル、および動作中プロトコルのうちの少なくとも１つを含む。

１６．　クレーム１５のＬＳＩ検証方法において、

前記所定のプロトコル制約条件を、ランダムに選択する。

１７．　クレーム１１または１４記載のＬＳＩ検証方法において、

前記制限措置は、シミュレーションの停止、実行速度低下および異常実行、シミュレーション結果の非出力、並びに、次ステップに渡すデータまたはキーの非生成のうちの少なくとも１つを含む。

１８．　ＬＳＩ検証方法は、

シミュレーションにおける不正アクセスをチェックするチェック回路を含む回路設計データを、暗号化処理するステップと、

暗号化された回路設計データを、シミュレーションによって検証するステップ

とを備え、

前記検証ステップは、前記チェック回路を動作させて、不正なアクセスによるシミュレーションを制限する。

１９．　クレーム１８のＬＳＩ検証方法において、

前記チェック回路は、シミュレーションにおいて、所定の制限情報のカウント値が上限値を超えたか否かをチェックする。

２０．　クレーム１８のＬＳＩ検証方法において、

前記チェック回路は、シミュレーションにおいて、プロトコル制約条件の違反の有無をチェックする。

２１．　ＬＳＩ設計方法は、

与えられた回路設計データから、タイミング情報を抽出するステップと、

前記回路設計データを、抽出したタイミング情報のみを合わせて、所定の変換ルールに従い、暗号化設計データに変換するとともに、少なくとも１つの論理ゲートにバッファを付加するステップと、

前記暗号化設計データについて、付加したバッファのサイズを調整するステップと、

バッファサイズ調整後の前記暗号化設計データを、前記所定の変換ルールをキーとして、復号化するステップとを備えている。

２２．　ＬＳＩ設計方法は、

固有ＩＤの判定回路とともに暗号化された回路設計データについて、復号化を行い、実際の設計データと、前記固有ＩＤの判定回路とを得るステップを備え、

前記ステップは、入力された固有パラメータによって、前記固有ＩＤの判定回路に正しい値を定義するステップを有する。

アブストラクト

ＬＳＩの設計において、暗号化処理を採り入れ、従来よりも回路設計データの機密性を高める。暗号化処理では、機密を必要とする回路の設計データに対して暗号化を行い、暗号化設計データおよび暗号化を解除するキーを生成する。暗号化設計データは設計・検証処理を実行する利用者に提供され、必要に応じてキーも併せて提供される。設計・検証処理では、暗号化設計データについて、元の回路の中身が秘匿されたまま各種の処理が行われる。復号化処理では、設計・検証処理が実行された後の暗号化設計データに対して復号化を行い、元の回路設計データを生成する。